

Penn Cambria Curriculum

Course Name	App and Game Development
Length of Course	1 credit / 1 days per week for two quarters
Grade Level	9-12
Prerequisites	None
Course Description	This course is designed to meet the needs of <i>all students</i> , not just those interested in computer careers. Prior knowledge of app and game development is not necessary. Students will learn to create interactive Android applications and Windows based and HTML based games using programs called App Inventor, GameMaker, and Kodu.
Units of Study	Learning App Inventor Learning GameMaker Learning Kodu
Materials	Text: N/A Supplemental Materials: Internet connection, Notepad

Standard Alignment:

PA Academic Standards for Business, Computer and Information Technology(2012)

- 15.4.12. A – Apply the creative and productive use of emerging technologies for educational and personal success.
- 15.4.12. G – Create an advanced digital project using sophisticated design and appropriate software/applications.
- 15.4.12. H – Use programming languages to develop logical thinking and problem solving skills.
- 15.4.12. J – Create a complex computer program to solve a problem.

PA Academic Standards for Reading and Writing in Science and Technology (draft 8/6/12)

- CC.3.5. 11-12. C – Follow precisely a complex multistep procedure when carrying out experiment, taking measurements, or performing technical tasks; analyze the specific results based on explanations in the text.
- CC.3.5. 11-12. D – Determine the meaning of symbols, key terms, and domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 11-12 texts and topics.
- CC.3.5. 11-12. G – Integrate and evaluate multiple sources of information presented in diverse formats and media (e.g., quantitative data, video, multimedia) in order to address a question or solve a problem.
- CC.3.6. 11-12. I – Write routinely over extended time frames (time for reflection and revision) and shorter time frames (a single sitting or a day or two) for a range of discipline-specific tasks, purposes, and audiences.

Unit: Learning App Inventor

Estimated Time: 30 days

Curricular Objectives:

- describe the features of a cell phone that make it a computer
- identify some of the types of apps available in the Android Marketplace
- explain the role of the component designer, block editor, and phone/emulator
- understand the role of problem decomposition - that it is important to break a larger problem into smaller parts and solve one part at a time
- understand the event-driven nature of App Inventor programming
- understand that variables contain values and make programs easier to modify later
- describe the role of procedures in making code more efficient and easy to understand
- describe several programming paradigms: event-driven, object-oriented, imperative, and functional

- explain how conditionals and loops allow control over what code is run
- access images and sounds to create an app
- access the viewer, components, and properties screens
- navigate the app inventor environment: designer, blocks editor, emulator and/or physical phone
- use the following app inventor components: accelerometer sensor, image, list-picker
- drag and touch on the **canvas** component for drawing
- control screen layout with **screen arrangement** components
- use event handlers that take arguments
- create global variables for remembering information while the app runs
- use camera component for app to access the phone's camera
- label the major parts of an App Inventor program
- understand how the components and blocks work and interact
- understand the difference between the user's experience and the program

Assessments/ Measurement of Objectives:

Daily projects/ work tasks

Cumulative project

Suggested Methods of Instruction / Learning Activities:

Practice exercises

Tutorial step by step directions

Direct instruction

Reading and Writing to Learn activities

Unit: Learning GameMaker

Estimated Time: 30 days

Curricular Objectives:

- Programming through a drag-n-drop method
- Programming by writing code using GML
- Follow technical and increasingly complex programming instructions in order and detail
- Program original game projects
- Use digital design resources and color theory to draw and animate sprites, objects, platforms, backgrounds and loops
- Become familiar and competent in using game engines (Game Maker); open files, save files, create and program original material, integrate separate files into a final game project, create and edit audio sound effects & music
- Technical writing; user instructions, game directions, game rules and document development process within a development team
- Demonstrate how to create a sprite, load a sprite, and make a sprite move
- Demonstrate how to add sound to a game and assign it to a sprite
- Demonstrate how to create different types of events
- Explain and Operate the different parts of the User Interface
- Demonstrate how to add background music to the game
- Demonstrate how to create natural motion for a moving sprite
- Demonstrate how to create variables and timers

- Demonstrate how to make a non-player character automatically generate an item based on randomly creating objects within a step event.

Assessments/ Measurement of Objectives:

Daily exercises

Cumulative project

Suggested Methods of Instruction / Learning Activities:

Practice exercises

Tutorial step by step directions

Direct instruction

Reading and Writing to Learn activities

Unit: Learning Kodu

Estimated Time: 30 days

Curricular Objectives:

- Understand the steps involved in creating a computer program
- Improve problem solving skills, and foster problem solving practices
- Follow online and offline directions more fluidly
- Compose stories in an alternative format and through varying mediums
- Implicitly practice math through branching and scoring
- Develop more positive attitudes towards computer programming
- Create increasingly complex games thus showing a deeper understanding for complex coding sequences
- Show evidence of perspective taking and empathy in game play
- Collaboratively work to create innovative solutions
- Navigate the Kodu macro environment using a keyboard and mouse
- Understand the foundational principles of programming
- Access the programming mode of Kodu, potentially adjusting simple code for a specific purpose
- Change and create a Kodu environment
- Use tiles for setting and the development of tone and mood of game world
- Use the keyboard to move characters in a game world
- Create paths on which characters will move
- Give objects behaviors
- Create a protagonist (user controlled), an antagonist (automated), and peripheral characters
- Base character behaviors and actions on environments and reactions to each other
- Begin to understand gaming plot and background story
- Be more attuned to strategy making
- Understand the influence of mood and tone on game play
- Understand cloning and creatables

Assessments/ Measurement of Objectives:

Daily exercises

Cumulative projects

Suggested Methods of Instruction / Learning Activities:

Practice exercises

Tutorial step by step directions

Direct instruction
Reading and Writing to Learn activities